

Walkera WK-0701

PCM Format

By Shaul Eizikovich

1. Introduction

This document describes the logic and frame of the Walkera WK-0701 R/C transmitter. This transmitter employs a special form of PCM that takes a few additional moments to understand.

I'm pleased to acknowledge [Eilert Johansson](#) for his important contribution to the understanding of this protocol.

2. Data description

Walkera WK-0701 R/C transmitter PCM includes eight channels. Each channel consists of 10 information bits. Not all channels are fully implemented. Some are only partially implemented while others are mere placeholders for future use.

In addition, there are six data validation fields: four Checksum fields and two Magic Number fields.

2.1. Channels:

All channels are defined by a 10-bit word. The most significant bit serves as a sign-bit while the 9 other bits signifies the absolute value of the channel. Therefore, the two extreme values of every channel are ± 511 while the mid-point (e.g. Centered stick) value is zero. Reversing a channel results in toggle of the channel's most significant bit. The following table reflects the **normal** mode.

<i>Channel</i>	<i>Mnemonic</i>	<i>Sign-bit</i>		<i>Notes</i>
		<i>0</i>	<i>1</i>	
1	Elevator	Push	Pull	
2	Ailerons	Left	Right	
3	Throttle	Push	Pull	<i>Flight mode</i> switch on position 'N'
4	Rudder	Left	Right	
5	Gear	Down	Up	Channel value is fixed at ± 375
6	Pitch			User cannot control this channel directly
7	Gyro			User cannot control this channel directly
8	Ch8			Placeholder for future use

2.2. Checksum fields:

There are four Checksum fields used by the receiver to validate the data. One pair corresponds to channels 1 to 4. The second corresponds to channels 5 to 8. The exact

algorithm for the calculation of these fields is closely related to the frame pattern and described later in this document.

2.3. Magic Number fields:

These are two fields holding rolling numbers, probably used by the receiver to detect lost frames.

There is a pair of Magic Number byte and a magic number bit. The pairs come in the following order, starting over again every four frames:

<i>Frame number</i>	<i>Magic Number Byte</i>	<i>Magic Number Bit</i>
4xN	0xF4	1
4xN+1	0x01	0
4xN+2	0x52	1
4xN+3	0xA3	0

3. Frame Description

3.1. General Structure

The frame consists of exactly 50 pulses. We shall mark the first pulse as P[1] and the last pulse as P[50]. The first pulse (P[1]) is high, so naturally all odd pulses (P[1], P[3] ... P[49]) will be high while all the even pulses (P[2], P[4] ... P[50]) will be low.

The first pulse (P[1]) is a long pulse (1306 μ S) marking the beginning of a new frame, this is the *sync* pulse. The following pulse (P[2]) carries one bit of information, P[3] carries 3 bits of information, and so on: All odd pulses carry 3 bits (octal data) and all even pulses carry 1 bit (binary data).



3.2. Octal data

The data carried with every odd data pulse (P[3] ... P[49]) is calculated from the length of the pulse according to the following table:

<i>Length</i>	<i>Value</i>
318 μ S	000
398 μ S	001
478 μ S	010
558 μ S	011
638 μ S	100
718 μ S	101
798 μ S	110
878 μ S	111

3.3. Binary data

The data carried with every even data pulse (P[2] ... P[50]) is calculated from the length of the pulse according to the following table:

<i>Length</i>	<i>Value</i>
288	0
438	1

3.4. Frame data

The best way to explain how to create or interpret the frame data will be using an example:

Pulse Number	2	3	4	5	6	7	8	9	10	11	12	13
Mnemonic	Bin1	Oct1	Bin2	Oct2	Bin3	Oct3	Bin4	Oct4	Bin5	Oct5	Bin6	Oct6
Length (μS)	288	478	288	798	438	318	288	558	438	718	438	478
Value	0	010	0	110	1	000	0	011	1	101	1	010

The above example covers only the first 12 data pulses (remember, the first pulse is the sync pulse). The value that these pulses carry is calculated simply by copying the data as it is: 001001101000001111011010 for this example.

3.5. Frame fields

Pulse	Width[μ S]	Mnemonic	Weight	Description
2	288/438	bin1		Elevator Sign bit – 0=Push 1=Pull
3	318-878	oct1	MSB (Elevator)	Elv[8]
				Elv[7]
				Elv[6]
4	288/438	bin2		Elv[5]
5	318-878	oct2		Elv[4]
				Elv[3]
				Elv[2]
6	288/438	bin3		Elv[1]
7	318-878	oct3	LSB (Elevator)	Elv[0]
				Ailerons Sign bit: - 0=Left, 1=Right
			MSB (Ailerons)	Ail[8]
8	288/438	bin4		Ail[7]
9	318-878	oct4		Ail[6]
				Ail[5]
				Ail[4]
10	288/438	bin5		Ail[3]
11	318-878	oct5		Ail[2]
				Ail[1]
			LSB (Ailerons)	Ail[0]
12	288/438	bin6		Throttle Sign bit – 0=Push 1=Pull (Switch Flight mode = N)
13	318-878	oct6	MSB (Throttle)	Thr[8]
				Thr[7]
				Thr[6]
14	288/438	bin7		Thr[5]
15	318-878	oct7		Thr[4]

Pulse	Width[μ S]	Mnemonic	Weight	Description
				Thr[3]
				Thr[2]
16	288/438	bin8		Thr[1]
17	318-878	oct8	LSB (Throttle)	Thr[0]
				Rudder Sign bit: - 0=Left, 1=Right
			MSB (Rudder)	Rud[8]
18	288/438	bin9		Rud[7]
19	318-878	oct9		Rud[6]
				Rud[5]
				Rud[4]
20	288/438	bin10		Rud[3]
21	318-878	oct10		Rud[2]
				Rud[1]
			LSB (Rudder)	Rud[0]
22	288/438	bin11	CS1	$\Sigma_{bin1-bin10} + \text{int}((\Sigma_{oct1-oct10})/8)$ OR 0x0001
23	318-878	oct11	CS2	$\Sigma_{oct1-oct10}$ OR 0x0007
24	288/438	bin12		Gear switch – 0=down 1=up
25	318-878	oct12	MSB (Gear)	Gear[8]
				Gear[7]
				Gear[6]
26	288/438	bin13		Gear[5]
27	318-878	oct13		Gear[4]
				Gear[3]
				Gear[2]
28	288/438	bin14		Gear[1]
29	318-878	oct14	LSB (Gear)	Gear[0]
				Pitch Sign bit
			MSB (Pitch)	Pch[8]
30	288/438	bin15		Pch[7]
31	318-878	oct15		Pch[6]
				Pch[5]
				Pch[4]
32	288/438	bin16		Pch[3]
33	318-878	oct16		Pch[2]
				Pch[1]
			LSB (Pitch)	Pch[0]
34	288/438	bin17		Gyro Sign bit
35	318-878	oct17	MSB (Gyro)	Gyr[8]

Pulse	Width[μ S]	Mnemonic	Weight	Description
				Gyr[7]
				Gyr[6]
36	288/438	bin18		Gyr[5]
				Gyr[4]
37	318-878	oct18		Gyr[3]
				Gyr[2]
38	288/438	bin19		Gyr[1]
			LSB (Gyro)	Gyr[0]
39	318-878	oct19		Channel8 Sign bit
			MSB (Channel8)	Ch8[8]
40	288/438	bin20		Ch8[7]
				Ch8[6]
41	318-878	oct20		Ch8[5]
				Ch8[4]
42	288/438	bin21		Ch8[3]
				Ch8[2]
43	318-878	oct21		Ch8[1]
			LSB (Channel8)	Ch8[0]
44	288/438	bin22	MSB (Magic)	Mgc[7]
				Mgc[6]
45	318-878	oct22		Mgc[5]
				Mgc[4]
46	288/438	bin23		Mgc[3]
				Mgc[2]
47	318-878	oct23		Mgc[1]
			LSB (Magic)	Mgc[0]
48	288/438	bin24	CS3	$\sum_{bin12-bin23} + \text{int}((\sum_{oct12-oct23})/8)$ OR 0x0001
49	318-878	oct24	CS4	$\sum_{oct12-oct23}$ OR 0x0007
50	288/438	bin25	Delimiter	Mgc[6]

Notes:

- All channels are calculated as 9-bit words preceded by a sign-bit.
- The meaning of the sign-bit differs according to the channel.
- For the first four channels (Elevator, Ailerons, Throttle and Rudder) – the value of the 9-bit word reflects the distance from the stick rest point (Mid-point). The sign-bit reflects the direction (Right/Left or Push/Pull).
- Gear Channel (CH5): Presently only pulse *bin12* is used for up/down.
- Magic Number¹ (P[47-44]): Rolling number in the following order: 0x94, 0x01, 0x52, 0x43

¹ These numbers may differ between different transmitters. Do not rely on them.

- Delimiter: Equal to (oct22 AND 0x001)
- Checksum values:
- CS1: Checksum for channels 1-4.
Calculation:
 - Shift the following sum (oct1+ ... +oct10) three bits to the right
 - Add it to the sum (bin1 + ... + bin10)
 - use only the lower bit
- CS2: Checksum for channels 1-4.
Calculation: Three Least Significant Bits of (oct1+ ... +oct10))
- CS3: Checksum for channels 5-8.
Calculation:
 - Shift the following sum (oct12+ ... +oct23) three bits to the right
 - Add it to the sum (bin12 + ... + bin23)
 - use only the lower bit
- CS4: Checksum for channels 5-8.
Calculation: Three Least Significant Bits of (oct12+ ... +oct23))